



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/049,211	02/05/2002	Siani Lynne Pearson	B-4487PCT 619499 -6	8087

22879 7590 11/24/2006

HEWLETT PACKARD COMPANY  
P O BOX 272400, 3404 E. HARMONY ROAD  
INTELLECTUAL PROPERTY ADMINISTRATION  
FORT COLLINS, CO 80527-2400

EXAMINER

SHERKAT, AREZOO

ART UNIT	PAPER NUMBER
----------	--------------

2131

DATE MAILED: 11/24/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

10/049,211

Applicant(s)

PEARSON ET AL.

Examiner

Arezoo Sherkat

Art Unit

2131

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 14 September 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-11, 19-22 and 26-47 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1, 2, 4-6, 8-11, 21, 22, 26, 28, 29 and 31 is/are rejected.
- 7) ☒ Claim(s) 3, 7, 19, 20, 27, 30, 32 and 34-47 is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 05 February 2002 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☒ All b) ☐ Some \* c) ☐ None of:
1. ☒ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)   | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date <u>6/19/2006</u> . | 6) <input type="checkbox"/> Other: _____  |

***Response to Amendment***

This office action is responsive to Applicant's amendment received on 9/14/2006. Claims 1 and 37 have been amended to correct the typographical errors. Claims 1-11, 19-22 and 26-47 remain pending.

***Response to Arguments***

Applicant's arguments filed 9/11/2006 have been fully considered but they are not persuasive.

Applicant argues, "To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings." MPEP §2142. Furthermore, "rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." In re Leonard Kahn, 04-1616, \*p. 15 (Fed. Cir., March 22, 2006) [emphasis added]. The Examiner's assertion of obviousness does not meet this requirement as it does not state where in the references themselves the skilled person can find the requisite motivation and, more importantly, the Examiner's allegation of motivation is self-contradictory in view of his assertion that Rabin teaches means for integrity checking the license-related code and preventing the license-related code from being loaded if the integrity check fails" (Remarks, page 15, first paragraph).

As previously communicated, Examiner emphasizes that Rabin et al., does disclose “means for integrity checking the license-related code (i.e., the tag) with reference to the signed version (i.e., signed tag)” (column 40, lines 47-54). Rabin et al., further discloses “and the public key certificate and preventing the license-related code from being loaded if the integrity check fails” (column 40, lines 55-67 and column 41, lines 9-17), where rejection means discarding, removing, or preventing the use of the instance of the software associated with the invalid tag). It also describes that the supervising program further verifies by use of the tag server’s public key (transferred by the tag server to the user’s device – Figure 2, element 116). Graunke et al. discloses a server 30 (i.e., tag server) **securely** distributing a conditional use private key (i.e., public key certificate) to a trusted entity (i.e., a legitimate user device) on a remote system by **wrapping the key in an executable tamper resistant key module** (Abstract).

Although Rabin et al. does infact check the integrity of the signed tag by the supervising program (column 40, lines 47-54); it does not expressly disclose a secure transmission of the public key certificate.

Therefore, it would have been obvious to one of ordinary skill at the time of invention to modify teachings of Rabin et al. with teachings of Graunke et al. to include secure distribution of a conditional use private key to a trusted entity on a remote system by wrapping the key in an executable tamper resistant key module. One of ordinary skill in the art would have motivated to modify the method of Rabin et al. to prevent the malicious user from trying to expose the key (i.e., public key certificate)(col. 5, lines 1-14).

Applicant argues, "Contrary to the Examiner's assertion, Rabin does not in fact teach means for integrity checking the license-related code and preventing the license-related code from being loaded if the integrity check fails".

Examiner contends that Rabin et al., does disclose "means for integrity checking the license-related code (i.e., the tag) with reference to the signed version (i.e., signed tag)" (column 40, lines 47-54). Rabin et al., further discloses "and the public key certificate and preventing the license-related code from being loaded if the integrity check fails" (column 40, lines 55-67 and column 41, lines 9-17), where rejection means discarding, removing, or preventing the use of the instance of the software and its associated tag).

Applicant argues, "Specifically, the Examiner asserts that Rabin teaches means storing license-related code comprising at least one of a secure executor for checking whether the platform or a user thereof is licensed to use particular data and for providing an interface for using the data and/or for monitoring its usage, and a secure loader for checking whether the platform or a user thereof is licensed to install particular data and/or for checking for data integrity before installation, by teaching the supervising program (SP) 209 which runs on the user's device (i.e. computer platform) to ensure that no unauthorized use of software takes place (abstract of Rabin). The Examiner then further asserts that Rabin teaches means for integrity checking the license-related code and preventing the license-related code from being loaded if the integrity check fails at col. 40 11.53-65, which discusses how the supervising program rejects an instance of software if an invalid tag is returned by a tag server - "where the Examiner

Art Unit: 2131

interprets rejecting the instance as preventing the code from being loaded if the tag fails the integrity check." Applicants respectfully point out that these two statements are contradictory - in the first statement the Examiner asserts that the supervising program corresponds to the claimed license-related code, and in the second statement the Examiner is clearly considering the software instance (i.e. the software product provided by the software vendor and the use of which is monitored by the supervising program) to correspond to the claimed license-related code. Thus, Applicants respectfully ask - which one is it? Claim 1 recites means for integrity checking and preventing the license-related code from being loaded if the integrity check fails".

Examiner responds that Rabin et al. teaches means for integrity checking the license-related code (i.e., supervising program verifies the integrity of the **signed tag** by computing a hash function value V and a hash function value U and comparing the two value. It then validates the signature of the signed tag using tag server's public key)(col. 40, lines 47-65) and preventing the license-related code from being loaded if the integrity check fails (i.e., "Rejection in step 254 can simply mean that the user device 104 **discards or removes or does not allow use** of the instance of software and **its associated tag** that were obtained in steps 250 and 251" – in both instances the signed tag is referenced and considered as the license-related code)(col. 41, lines 9-17).

Applicant argues, " there is absolutely nothing in Rabin that discusses means for preventing the supervising program from being loaded in some sort of check fails; in fact, there is no discussion of any sort of checks performed on the supervising program

Art Unit: 2131

whatsoever. And, of course, there is no discussion anywhere in Rabin of the software products monitored by the supervising program having any of the claimed limitations of the license-related code”.

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., preventing the supervising program (i.e., **executor**) from being loaded in some sort of check fails) are not recited in the rejected claim(s)(i.e., the license-related code corresponds to an instance of software and its associated tag). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

As emphasized earlier, Rabin et al., does disclose **discarding or removing or not allowing use** of the instance of software and **its associated tag**, if the hash value does not properly evaluate (i.e., corresponding to the license-related code claimed in the instant application)(col. 41, lines 3-17).

Applicant argues, “there is in fact not only no motivation to combine Barber with Sigbjornsen as alleged by the Examiner, but that the references actually teach against this very combination”.

Examiner responds that “Sigbjornsen et al. teach software licenses stored in trusted modules (smart cards)(column 7, lines 31-34). Sigbjornsen et al. further teach that smart cards provide flexibility (column 7, lines 29) and are considered the most tamper-proof protection of data (column 8, line 61 – column 9, line 7). It would have been obvious to one of ordinary skill in the art at the time of invention to store the

Art Unit: 2131

licenses of Barber et al. in the smart cards of Sigbjornsen et al. to protect them from tampering". Moreover, Sigbjornsen et al. specifically discloses its software protection system in different types of data networks to which the computer may establish a connection such as WAN, LAN, and in particular Internet (col. 5, lines 25-53).

### ***Claim Objections***

Claims 1 and 29 are objected to because of the following informalities:

The limitation "means storing" should read as "means for storing". Appropriate correction is required.

### ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-2, 4-6, 8-11, 21-22, 26, 29, 31, and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rabin et al., US Patent 6,697,948 B1, in view of Graunke et al., U.S. Patent 5,991,399. Examiner notes that corresponding prior art terms may accompany the claim language in bracketed form.



Regarding claim 1, Rabin et al. teach a computer platform having:  
means storing license-related code (i.e., an instance of software and its associated tag) comprising at least one of:

a secure executor (SP) for checking whether the platform or a user thereof is licensed to use particular data and for providing an interface for using the data (figure 4, item 209, figure 8, column 47, lines 14-45, column 40, lines 6-9);

means storing a hashed version of the license-related code signed with the third party's private key (column 40, lines 47-65, where the signed tag is stored on the user device, showing that the device provides means for storing signed license-related code);

means for integrity checking the license-related code with reference to the signed version and the public key certificate and preventing the license-related code from being loaded if the integrity check fails (column 40, lines 53-65, where Examiner interprets rejecting the instance as preventing the code from being loaded if the tag fails the integrity check).

Rabin et al. do not teach a trusted module.

Graunke et al. teach a trusted module (tamper resistant key module) which is resistant to internal tampering (column 7, line 31) and which stores a third party's public key certificate (column 7, lines 31-58), and that this key is used to verify the integrity and authenticity of the license-related code (trusted player) (column 8, lines 39-60). Graunke et al. further provide the motivation that a key module verifies the authenticity of a software executor (storage device reader) and that access to the content is allowed

Art Unit: 2131

and that making it tamper resistant ensures that an attacker will not be able to modify the integrity parameters or otherwise alter the key module (column 4, lines 48-50, lines 64-67, column 5, lines 1-2, lines 11-14). It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to use the tamper-resistant module (key module) of Graunke et al. with the platform of Rabin et al. to ensure the authenticity of the executor and that access to the content is allowed.

Regarding claim 29, Rabin et al. teach a computer platform having:

means storing license-related code (supervising program (SP), tag table, and optional fingerprint table, see figure 4, items 209, 126, and 210) comprising, for at least one group of data (instances of software), a respective software executor which specifies the respective group of data and which is operable to act as an interface to the group of data, the license related code further comprising at least one of:

a secure executor (SP) for checking whether the platform or a user thereof is licensed to use particular data and for providing an interface for using the data (figure 4, item 209, figure 8, column 47, lines 14-45, column 40, lines 6-9);

means storing a hashed version of the license-related code signed with the third party's private key (column 40, lines 47-65, where the signed tag is stored on the user device, showing that the device provides means for storing signed license-related code);

wherein the computer platform is programmed so that, upon booting of the platform (column 60, lines 17-33):

the license-related code is integrity checked with reference to the signed version and the public key certificate (column 40, lines 53-65, and column 60, lines 17-33, where Examiner believes it would have been obvious to use a signature with the fingerprint in order to produce a signed digest for verification, as was done to verify the integrity of a signed tag in column 40, lines 53-65); and

if the integrity check fails, the license-related code is prevented from being loaded (column 40, lines 53-65, where Examiner interprets rejecting the instance as preventing the code from being loaded if the tag fails the integrity check, and column 60, lines 17-33, where the example checks the operating system, but states that it can also be used to check the SP); and

The platform of Rabin et al. does not teach a trusted module.

Graunke et al. teach a trusted module (tamper resistant key module) which is resistant to internal tampering (column 7, line 31) and which stores a third party's public key certificate (column 7, lines 31-58), that this key is used to verify the integrity and authenticity of the license-related code (trusted player) (column 8, lines 39-60), and that part of the license related-code (IVK, synonymous to the device which checks integrity and authenticity of SP and OS in Rabin et al.) is stored in the trusted module (column 7, lines 31-40). Graunke et al. further provide the motivation that a key module verifies the authenticity of a software executor (storage device reader) and that access to the content is allowed and that making it tamper resistant ensures that an attacker will not be able to modify the integrity parameters or otherwise alter the key module (column 4, lines 48-50, lines 64-67, column 5, lines 1-2, lines 11-14). It would have been obvious to

Art Unit: 2131

a person of ordinary skill in the art at the time of applicant's invention to use the tamper-resistant module (key module) of Graunke et al. in the platform of Rabin et al. to ensure the authenticity of the executor and that access to the content is allowed.

As per claim 2, the platform of Rabin et al. and Graunke et al. teaches the platform of claim, wherein the means for integrity checking further comprises:

means for reading and hashing the license-related code to produce a first hash;

means for reading and decrypting the signed version using the public key certificate to produce a second hash; and

means for comparing the first and second hashes (Rabin et al., column 40, lines 47-61);

As per claim 4, the platform of Rabin et al. and Graunke et al. teaches the platform of claim 1, wherein the license-related code (trusted player) also includes a library of interface subroutines which can be called in order to communicate with the trusted module (Graunke et al., column 8, lines 34-35, column 8, line 61 – column 9, line 1, where the license-related code executes the trusted module and play content decrypted by the module, therefor showing that it contains libraries to communicate with it).

As per claim 5, the platform of Rabin et al. and Graunke et al. teaches the platform of claim 1, wherein the license-related code includes, for at least one group of

Art Unit: 2131

data (software instance), a software executor which specifies the respective group of data and which is operable to act as an interface to that group of data (Rabin et al., figure 4, item 209, figure 8, column 47, lines 14-45, column 40, lines 6-9).

As per claim 6, the platform of Rabin et al. and Graunke et al. teaches the platform of claim 1, wherein the means storing the license-related code and/or the means storing the hashed version of the license-related code are provided, at least in part, by the trusted module (Graunke et al., column 7, lines 31-40, where the IVK of Graunke et al. is synonymous to the device which checks integrity and authenticity of SP and OS in Rabin et al.).

As per claims 8-10, the platform of Rabin et al. and Graunke et al. teaches the platform of claim 1 wherein:

the operating system is operable to request the secure loader to license-check whether the platform or a user thereof is licensed to install that particular data and/or to check the integrity of that data;

in response to such a request, the secure loader is operable to perform such a check and respond to the operating system with the result of the check; and

in dependence upon the response, the operating system is operable to install or not to install the particular data (Rabin et al., column 60, lines 55-61, where the check may be done prior to installation, as described above in reference to execution).

As per claim 11, the platform of Rabin et al. and Graunke et al. teaches the platform of claim 10, wherein the license-related code includes, for at least one group of data (software instance), a software executor which specifies the respective group of data and which is operable to act as an interface to that group of data (Rabin et al., figure 4, item 209, figure 8, column 47, lines 14-45, column 40, lines 6-9).

As per claim 21, the platform of Rabin et al. and Graunke et al. teaches the platform of claim 1, wherein:

the secure executor contains at least one licensing model (Rabin et al., column 59, lines 37-57);

the operating system is operable to request the secure executor that particular data be used (figure 8, column 47, lines 14-33);

in response to such a request, the secure executor is operable:

to perform a license-check using the, or one of the, licensing models; and

upon successful license-check, to request the operating system to use the data (Rabin et al., column 59, lines 37-57, column 47, lines 14-33).

As per claim 22, the platform of Rabin et al. and Graunke et al. teaches the platform of claim 21, wherein the operating system is programmed to use the particular data only in response to the secure executor or the software executor (Rabin et al., column 59, lines 37-57, column 47, lines 14-33).

As per claim 26, the platform of Rabin et al. and Graunke et al. teaches the platform of claim 21, wherein the trusted module is operable to log the request to the operating system to use the data (Rabin et al., column 31, lines 57-64, column 42, lines 48-58).

As per claim 31, the platform of Rabin et al. and Graunke et al. teaches the platform of claim 29, wherein the operating system is programmed to install (play) the particular data only response to the trusted module (Graunke et al., column 8, line 61 – column 9, line 1).

As per claim 33, the platform of Rabin et al. and Graunke et al. teaches the platform of claim 29, wherein if the check succeeds, the trusted module is operable to generate a log for auditing the particular data (Rabin et al., column 31, lines 57-64, column 42, lines 48-58).

Claim 28 is rejected under 35 U.S.C. 103(a) as being unpatentable over Barber et al., US Patent 5,390,297, in view of Sigbjornsen et al., US Patent 6,266,416.

Regarding claim 28, Barber et al. teach a method of transferring a license (or a key therefor) for data from a first computer platform to a second computer platform, comprising the steps of:

Art Unit: 2131

setting up secure communication between the platforms (column 7, lines 58-64, claim 2);

sending the license or the key therefor from the first platform (second node) to the second platform (local node) using the secure communication (); and

deleting the license or key therefor from the first platform (second node) (figure 3, column 10, line 65 – column 11, line 15).

The method of Barber et al. does not teach trusted modules containing the license.

Sigbjornsen et al. teach software licenses stored in trusted modules (smart cards) (column 7, lines 31-34). Sigbjornsen et al. further teach that smart cards provide flexibility (column 7, lines 29) and are considered the most tamper-proof protection of data (column 8, line 61 – column 9, line 7). It would have been obvious to one of ordinary skill in the art at the time of invention to store the licenses of Barber et al. in the smart cards of Sigbjornsen et al. to protect them from tampering.

### ***Allowable Subject Matter***

Claims 3, 7, 19, 20, 27, 30, 32, 34-47 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

### ***Conclusion***



**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

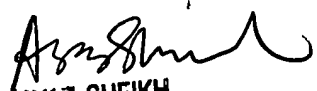
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Arezoo Sherkat whose telephone number is (571) 272-3796. The examiner can normally be reached on 8:00-4:30 Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on (571) 272-3795. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2131

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

A.S.  
Patent Examiner  
Group 2131  
November 16, 2006

  
AYAZ SHEIKH  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100